

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Aplikasi Mobile**

Menurut Buyens (2001) aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ketempat yang lain. Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* menurut Rangsang Purnama ( 2010 ) adalah sebutan untuk aplikasi yang berjalan di *mobile device* . Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya.

Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemai oleh pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita mejadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

Perangkat *mobile* memiliki banyak jenis dalam hal ukuran,desai layout, tetapi mereka memiliki kesamaan karakteristik yang sanagt berbeda dari dekstop sistem. Perangkat *mobile* memiliki memory yang kecil.

#### **2.2 Pengertian Les Private**

Les adalah pelajaran tambahan di luar jam sekolah, sedangkan privat berarti pribadi, tersendiri. Berdasarkan definisi tersebut maka pengajar les privat seseorang yang mengajar atau memberi bimbingan pelajaran tambahan pada mata pelajaran tertentu di luar jam belajar sekolah yang diadakan secara pribadi di rumah bagi siswa SD. Mata pelajaran yang disediakan les private ada mata pelajaran umum yaitu matematika, Bahasa Indonesia, IPS, IPA, PKN, dan pelajaran Bahasa Inggris ,TPA yaitu mata pelajaran Agama. Jasa yang diberikan oleh guru les berupa waktu dan intelektual. Guru les memberikan bimbingan, pemahaman, pengertian, penjelasan, dan aspek – aspek kogintif yang lain kepada siswa les. Selanjutnya jasa tadi akan dihargai oleh orangtua siswa dengan sejumlah biaya sesuai dengan

keepakatan sebelumnya.

### **2.3 Penjadwalan**

Penjadwalan merupakan proses perencanaan pengalokasian sumber daya dalam mengerjakan aktifitas job pada suatu waktu tertentu, yang kemudian akan dilakukan pengurutan kerja pada setiap pusat pemrosesan sehingga dicapai optimalitas dalam pengerjaannya. Berikut adalah pengertian penjadwalan menurut beberapa para ahli :

1. Menurut Baker and Trietsch (2013), Penjadwalan merupakan pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dialokasikan untuk dikerjakan pada beberapa mesin dalam jangka waktu tertentu.
2. Menurut Pinedo (2016), Penjadwalan adalah suatu fungsi dalam pengambilan keputusan yang biasa digunakan oleh perusahaan manufaktur atau jasa. Yang berkaitan dengan alokasi sumber daya untuk mengerjakan job selama waktu tertentu yang memiliki tujuan pengoptimalan.
3. Menurut Ginting (2009), Penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin.

Penjadwalan melibatkan elemen dasar yaitu job atau operasi. Disetiap operasi dibutuhkan alokasi tenaga operator, mesin peralatan produksi, dan dari aspek lainnya. Sehingga fungsi penjadwalan produksi akan sangat membantu untuk pengambilan keputusan dalam proses kelangsungan produksi agar lebih efisien.

### **2.4 Algoritma *Waiting Line***

Menurut Heizer dan Render (2017) menyatakan bahwa “antrian (*waiting line/queue*) ialah item-item atau orang-orang dalam suatu baris yang menunggu dilayani”.

Menurut Siswanto (2007) tujuan dasar dari model-model antrian adalah meminimumkan sekaligus dua jenis biaya, yaitu biaya langsung untuk menyediakan pelayanan dan biaya individu yang menunggu untuk memperoleh pelayanan. Perbedaan antara jumlah permintaan terhadap fasilitas pelayanan dan kemampuan fasilitas untuk melayani menimbulkan dua konsekuensi logis, yaitu timbulnya

antrian dan timbulnya pengangguran kapasitas. Antrian yang panjang karena kemampuan fasilitas pelayanan lebih rendah dari jumlah pemakainya, jelas akan memunculkan garis tunggu sehingga mereka yang antri atau ada digaris tunggu itu akan menanggung opportunity cost. Se jauh opportunity cost itu negative, maka mereka mungkin bersedia untuk tetap digaris tunggu. Namun sebaliknya mereka pasti akan keluar dari garis tunggu dan itu berarti kerugian. Di sisi lain, penyediaan kapasitas pelayan yang terlalu berlebihan sehingga tingkat penggunaan fasilitas tersebut rendah, jelas akan menaikkan biaya setiap rata-rata. Oleh karena itu, kedua jenis biaya tersebut harus diminimumkan.

Menurut Heizer dan Render (2017) menyatakan bahwa terdapat empat struktur dasar sistem antrian, terdiri atas :

- a. Sistem antrian jalur tunggal (Single Channel Single Phase) : Sebuah sistem pelayanan yang memiliki satu jalur dan satu titik pelayanan.
- b. Sistem antrian jalur berganda (Single Channel Multi Phase) : Sebuah sistem pelayanan yang memiliki satu jalur dengan beberapa titik pelayanan.
- c. Sistem antrian satu tahap (Multi Channel Single Phase): Sebuah sistem dimana pelanggan menerima pelayanan hanya dari satu stasiun dan kemudian pergi meninggalkan sistem.

Menurut Heizer dan Render (2015) menyatakan bahwa ada empat model antrian namun pada penelitian ini menggunakan model *Multiple Channel Model (M/M/s)* Menurut Heizer dan Render (2017) pada *Multiple Channel Model* ini terdapat dua atau lebih jalur atau stasiun pelayanan yang tersedia untuk melayani pelanggan yang datang. Asumsinya bahwa pelanggan yang menunggu pelayanan membentuk satu jalur yang akan dilayani pada stasiun pelayanan yang tersedia pertama kali pada saat itu. Model ini juga mengasumsikan bahwa pola kedatangan mengikuti distribusi eksponensial negatif. Pelayanan dilakukan secara FCFS, dan semua stasiun pelayanan diasumsikan memiliki tingkat pelayanan yang sama.

Pola kedatangannya berdistribusi Poisson dan pelayanan berdistribusi Eksponensial. Kedatangan dilayani dengan disiplin antrian FIFO. M/M/S adalah model antrian dengan pelayanan jamak, yaitu, konsumen tidak perlu menunggu lama karena tidak hanya satu server yang terbuka tetapi terdapat dua atau lebih server yang melayani

Rumus untuk model ini adalah sebagai berikut:

| Rumus   | Keterangan  | Satuan    |
|---|---|-----------|
| $P_0 = \frac{1}{\left[ \sum_{n=0}^{M-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \right] + \frac{1}{M!} \left(\frac{\lambda}{\mu}\right)^M \frac{M\mu}{M\mu-\lambda}}$ | Probabilitas 0 unit dalam system (system menganggur)  |           |
| $L_s = \frac{\lambda\mu\left(\frac{\lambda}{\mu}\right)^M}{(M-1)!(M\mu-\lambda)^2} P_0 + \frac{\lambda}{\mu}$   | Jumlah rata-rata unit (konsumen) didalam system (tunggu dan akan dilayani)                  | Pelanggan |
| $W_s = \frac{L_s}{\lambda}$   | Waktu rata-rata unit yang dihabiskan didalam system (waktu tunggu ditambah waktu pelayanan) | Menit     |
| $L_q = L_s - \frac{\lambda}{\mu}$   | Jumlah rata-rata unit yang menunggu didalam antrian   | Pelanggan |
| $W_q = W_s - \frac{1}{\mu}$   | Waktu rata-rata unit yang dihabiskan untuk menunggu didalam antrian                         | Menit     |

Sumber: Heizer dan Render (2017)

**Gambar 2. 1** Rumus Algoritma Waiting Line

Keterangan :

M : Jumlah jalur yang terbuka

$\lambda$  : Jumlah kedatangan rata-rata per satuan waktu

$\mu$  : Jumlah orang yang dilayani per satuan waktu

$P_0$  : Probabilitas terdapat 0 orang dalam sistem

$L_s$  : Jumlah pelanggan rata-rata dalam sistem

$W_s$ : Jumlah waktu rata-rata yang dihabiskan seorang pelanggan dalam antrian atau sedang dilayani (dalam sistem)

$L_q$  : Jumlah orang atau unit rata-rata yang menunggu dalam antrian

$W_q$ : Waktu rata-rata yang dihabiskan oleh seorang pelanggan atau unit untuk menunggu dalam antrian.

## 2.5 PHP

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah web-server (serverside). PHP diciptakan oleh programmer unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus September 1994. Script PHP adalah bahasa program yang berjalan pada sebuah web server, atau sering disebut serverside. Oleh karena itu, PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain,

yaitu mengolah data dengantipe apapun, menciptakan halaman web yang dinamis,serta menerima dan menciptakan cookies, dan bahkan PHP bisa melakukan lebih dari itu (Harison, 2016) [4]

## 2.6 Xampp

Xampp merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Xampp merupakan kepanjangan dari hurufnya yaitu :

Andriyani dan Siyoperman Gea, Sistem Monitoring Peralatan Bengkel

X : Program ini dapat dijalankan dibanyak sistem operasi, seperti Windows, Linux, Mac OS dan juga Solaris.

A : Apache, merupakan aplikasi web server. Tugas utama dari Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat web atau user.

M : MySQL, merupakan aplikasi data server. Perkembangannya disebut juga Sql yang merupakan kepanjangan dari Structured Query Language. Sql merupakan bahasa terstruktur yang digunakan untuk mengolah database.

P : PHP, merupakan bahasa pemrograman web, dimana user dapat menggunakan bahasa pemrograman ini untuk membuat web yang bersifat server-side scripting.

P : Perl, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh Larry Wall di mesin Unix. (Andriyani, 2016)[2]



**Gambar 2. 2** Xampp

Sumber: (<https://www.nesabamedia.com/pengertian-xampp/>)

## 2.7 MySQL

Beberapa pengertian MySQL menurut para ahli:

1. Menurut Raharjo (2011:21), “MySQL merupakan RDBMS (atau *server database*) yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*”.
2. Menurut Kadir (2008:2), “MySQL adalah sebuah software *open source* yang digunakan untuk membuat sebuah *database*.”

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa *MySQL* adalah suatu software atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*.

## 2.8 Java

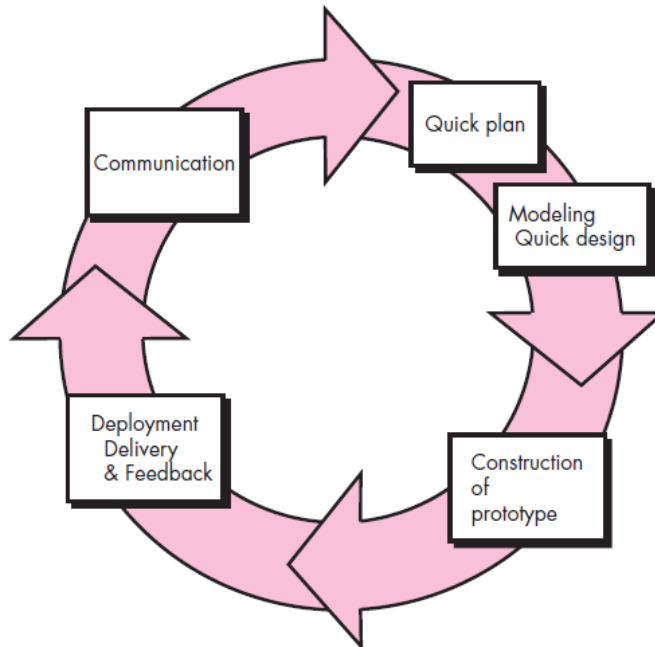
Java merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet/ jaringan komunikasi. Melalui teknologi java, dimungkinkan perangkat audio stereo dirumah terhubung jaringan komputer (Haryanto, 2011) . Java tidak lagi hanya untuk membuat applet yang memerintah halaman web tapi java telah menjadi bahasa untuk pengembangan aplikasi skala interprise berbasis jaringan besar.

## 2.9 Metode Pengembangan Sistem

Metodelogi yang digunakan untuk membangun sistem ini adalah Model Prototype. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematis, dengan beberapa tahapan, yaitu: System Engineering, Analysis, Design, Coding, Testing dan Maintenance.

Pressman (2010) mengutarakan bahwa prototype adalah pengembangan yang cepat dan pengujian terhadap model kerja (prototipe) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. Prototype disebut juga desain aplikasi cepat (rapid application design/RAD) karena menyederhanakan dan mempercepat desain sistem. Bagian user kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis

dengan memahami kebutuhan user dan menerjemahkannya ke dalam bentuk model (prototipe). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan user. Model prototype dapat dilihat pada gambar 2.3 berikut.



**Gambar 2. 3 Model Prototype**

Penjelasan dari gambar 2.3 diatas adalah sebagai berikut :

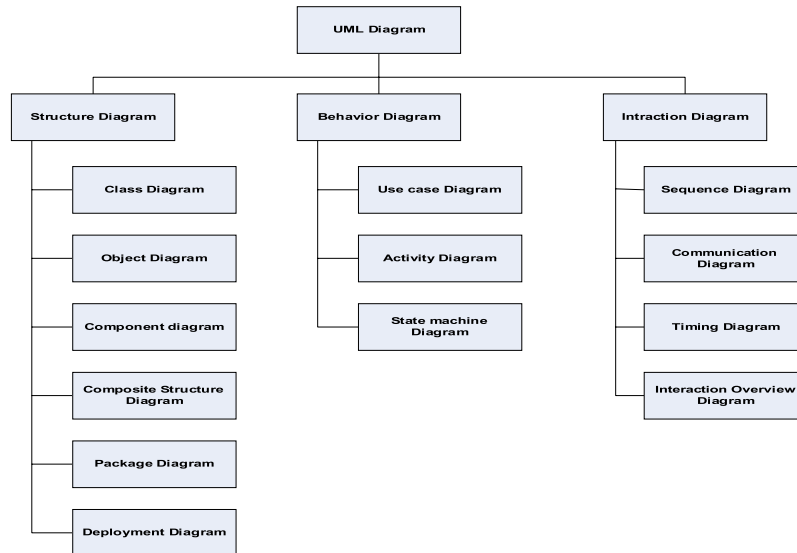
Tahap pertama adalah communication dan pengumpulan data awal yaitu tahap suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya. Tahap kedua adalah quick plan yaitu analisis terhadap kebutuhan pengguna. Tahap ketiga adalah modelling quick design yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Tahap keempat adalah construction of prototype adalah pembuatan perangkat prototype termasuk pengujian dan penyempurnaan. Tahap kelima adalah deployment, delivery, and feedback adalah tahap penyerahan sistem ke pengguna dan umpan balik.

## **2.10 Perancangan Sistem UML (*Unified Modelling Language*)**

Menurut Rosa A.S dan M. Shalauddin (2011) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia

industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



**Gambar 2. 4** Diagram UML (Unified Modelling Language)

Berikut ini merupakan penjelasan singkat dari pembagian kategori tersebut:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsystem pada suatu sistem.

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modeling Language*):

1. *Class Diagram*

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut maupun metode atau operasi. Atribut merupakan variable-



variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2. 1 Simbol-simbol Class Diagram**

| <b>Simbol</b>   | <b>Deskripsi</b>  |          |            |                            |
|---|---|----------|------------|----------------------------|
| <p>Kelas</p> <table border="1" style="margin-left: 20px;"> <tr> <td><b>nama_kelas</b></td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table> | <b>nama_kelas</b>   | +atribut | +operasi() | Kelas pada struktur sistem |
| <b>nama_kelas</b>   |   |          |            |                            |
| +atribut  |   |          |            |                            |
| +operasi()  |   |          |            |                            |
| <p>Antarmuka/<i>Interface</i></p> <p style="text-align: center;">○</p> <p><b>nama_interface</b></p>   | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek  |          |            |                            |
| <p>Asosiasi/<i>asociation</i></p> <p style="text-align: center;">—————</p>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>  |          |            |                            |
| <p>Asosiasi berarah/<i>directed association</i></p> <p style="text-align: center;">—————→</p>   | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> . |          |            |                            |
| <p>Generalisasi</p> <p style="text-align: center;">    ▷</p>  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)   |          |            |                            |
| <p>Kebergantungan/<i>dependecy</i></p> <p style="text-align: center;">.....→</p>  | Relasi antar kelas dengan makna kebergantungan antar kelas  |          |            |                            |
| <p>Agregasi/<i>agregation</i></p> <p style="text-align: center;">◊</p>  | Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )  |          |            |                            |

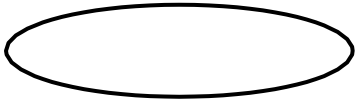


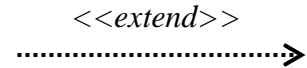
Sumber: Rosa, A.S (2011:123-124)



## 2. Use Case Diagram

*Use Case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2. 2 Simbol-Simbol Use Case Diagram**

| Simbol  | Deskripsi  |
|---|--|
| <p><i>Use Case</i></p>               | <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>  |
| <p>Aktor/<i>actor</i></p>          | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang</p>   |
| <p>Asosiasi/<i>asociation</i></p>  | <p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>   |
| <p>Ekstensi/<i>extend</i></p>      | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i></p> |



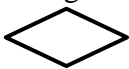
|   |  |
|---|--|
|   | tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan   |
| Generalisasi/ <i>generalization</i><br><br>                      | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya  |
| Menggunakan/ <i>Include/uses</i><br><br><< <i>include</i> >><br> | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. |



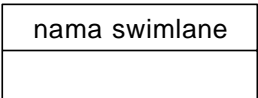
Sumber: Rosa, A.S (2011)

### 3. Activity Diagram

*Activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

**Tabel 2. 3 Simbol-simbol Activity Diagram**

| Simbol  | Deskripsi  |
|---|--|
| Status awal<br><br>                  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| Aktivitas<br><br>                    | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja      |
| Percabangan/ <i>decision</i><br><br> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu             |

|  |  |
|--|--|
|  |  |
| Penggabungan/ <i>join</i><br> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu            |
| Status akhir<br>              | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i><br>           | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi        |

Sumber: Rosa, A.S (2011)

## 2.11 Penelitian Terkait

Berikut ini merupakan penelitian terkait yang digunakan oleh peneliti sebagai bahan referensi adalah sebagai berikut :

| No | Nama Peneliti                      | Judul   |   |
|----|------------------------------------|---|---|
| 1  | Dinar Ajeng Kristiyanti (2018)[14] | <b>PENERAPAN METODE WAITING LINE UNTUK EVALUASI PELAYANAN PENJUALAN MERCHANDISE KAMPUS PADA PT. COME INDONUSA JAKARTA</b> | Penelitian ini dibuat karena manajemen operasional komputer yang digunakan belum berlangsung dengan optimal, terdeteksi terdapat antrian dalam pelayanan pelanggan yang kurang efektif dan efisien. Setelah dilakukan uji coba melalui metode |

|    |                              |  |   |
|----|------------------------------|--|---|
|    |                              |  | Waiting Line  |
| 2. | Sumiah Nasution              | <b>Penerapan Aplikasi pada Rumah Dakwa Mulia Privat dalam Dunia Pendidikan</b>               | embaga kursus bahasa Arab di kota Bekasi, yaitu Rumah Dakwah Privat Mulia. Saat ini, penyelenggaraan Rumah Dakwah Mulia masih menganut sistem manual. Oleh karena itu, penulis berupaya memberikan solusi dengan mengembangkan dan menerapkan alur kerja sistem administrasi yang mampu mengolah data secara cepat dan akurat.      |
| 3  | Relo Gagah Pambudi (2018)[2] | <b>APLIKASI MONITORING DAN MEDIA PEMBELAJARAN BAGASTA LES MUSIK JOMBANG BERBASIS ANDROID</b> | Aplikasi ini dibuat karena Penelitian ini bertujuan untuk mengembangkan teknologi informasi di bidang musik, pada Bagasta Les Musik Jombang yang belum mempunyai sebuah sistem monitoring yang dapat di akses oleh wali murid untuk mendapat informasi perkembangan dari murid, tanpa harus menggunakan sistem lama dengan mencatat |

|  |  |  |  |
|--|--|--|--|
|  |  |  | semua perkembangan<br>murid pada buku<br>pendamping di<br>karenakan kurang efisien |
|--|--|--|--|

### 2.12 Blackbox Testing

Black box testing merupakan teknik pengujian yang melakukan pendekatan kebutuhan dasar program dalam mengecek fungsional program (Swastika & Putra, 2016:73). Menurut Sukanto & Shalahuddin (2015:275) mendefinisikan bahwa “black box testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”. Berdasarkan penjelasan diatas mengenai Pengujian unit (black box testing) dapat disimpulkan bahwa black box testing adalah suatu pengujian program yang dibuat bertujuan untuk menemukan kesalahan yang tidak diduga sertamengatasi penyelesaian penemuan kesalahan yang terjadi.